SocACL: An ASP-Based Access Control Language for Online Social Networks

Edward Caprin and Yan Zhang

Artificial Intelligence Research Group
School of Computing, Engineering and Mathematics
University of Western Sydney, Kingswood, Australia
{e.caprin,y.zhang}@uws.edu.au

Abstract. *Online Social Networks* (OSNs), such as Facebook, encourage their users to disclose significant amounts of personal information to facilitate connecting and sharing content with other users. This has resulted in some OSNs holding vast amounts of information about their users; all of which is readily available via their profile page. As such, OSNs are particularly vulnerable to privacy breach attacks. With the impact these breaches varying from simply embarrassing the user, to negatively influencing the decision of a potential employer, identity theft and even physical harm it is important that these breaches are addressed. In this research we approach privacy management in OSNs as an access control problem, proposing a fine-grained, formal *Attribute-Based Access Control* (ABAC) language; *SocACL* (Social Access Control Language). SocACL is based on *Answer Set Programming* (ASP) and allows for policy specification using the most abundant sources of information available in OSNs; user attributes and relationships.

Keywords: Answer Set Programming, Online Social Networks, privacy, access control, Attribute-Based Access Control.

1 Introduction

Online Social Networks (OSNs), such as Facebook and LinkedIn, encourage their users to disclose significant amounts of personal information to facilitate connecting and sharing content with other users. This has resulted in some OSNs holding vast amounts of information about their users; all of which is readily available via their profile page. As such, OSNs are particularly vulnerable to privacy breach attacks [3]. With the impact these breaches varying from simply embarrassing the user, to negatively influencing the decision of a potential employer, identity theft and even physical harm it is important that these breaches are addressed. OSN operators have responded to privacy concerns by providing user customisable privacy settings. However, these have proven ineffective, often resulting in settings that do not reflect the intentions of the user [5]. This is in part due to the coarse-grained nature of the information on which these settings are based. In this research we approach privacy management in OSNs

B. De Decker et al. (Eds.): CMS 2013, LNCS 8099, pp. 207–210, 2013.

as an access control problem, proposing a fine-grained, formal *Attribute-Based Access Control* (ABAC) language; *SocACL* (Social Access Control Language). SocACL is based on *Answer Set Programming* (ASP) and allows for policy specification using the most abundant sources of information available in OSNs; user attributes and relationships.

2 Answer Set Programming (ASP)

The semantics of SocACL is defined as a translation to ASP. ASP is a form of declarative programming well suited to representing domain specific knowledge [1], making it ideal for capturing the wide range of features found in OSNs. An ASP program is a finite set of rules that describes some set of knowledge and are used with inference engines, such as DLV [4], to generate sets of conclusions that can be inferred from the program called *answer sets*, on which we base SocACL's policy evaluation system.

3 SocACL EBNF

```
      Query = NAME 'asks' NAME · ACT · OBJ · PURPOSE '?'
     Policy = {NAME 'says' (Rule | Definition) ';'}
       Rule = Head ['if' Body]
 Definition = Def-Obli | Def-RelC | Def-Desc
       Head = Auth | Attr ':' SF · PIF | Rel-Dir ':' SF | Dele
       Body = ( ['not'] [Prin 'says'] BTerm | Aggr | Cons )[',' Body]
      BTerm = Attr | Desc | Rel-Dir | Rel-Sind | Rel-Rind
       Auth = ('allow' | 'deny') · Prin · ACT · OBJ · PURPOSE · OBLI-NAME
       Attr = Prin · ATTR-NAME [ {·Val} ]
   Def-Obli = 'define' · 'obligation' · OBLI-NAME · ACT · Prin · NUM
   Def-RelC = 'define' · 'relchain' · RELCHAIN-NAME · '('Body')'
  Def-Desc = 'define' · 'description' · DESC-NAME · VAR · '('Body')'
       Aggr = VAR '=' Aggr-Op · VAR · '('Body')' | Aggr-Op · VAR · '('Body')' · Aggr-Cmp
   Aggr-Cmp = ('exactly' | 'atleast' | 'atmost') · Val | 'between' · Val · Val
   Aggr-Op = 'count' | 'sum' | 'min' | 'max'
       Desc = SUB · 'description' · DESC-NAME
    Rel-Dir = SUB · 'relationship' · REL-TYPE · SUB
   Rel-Sind = SUB · 'sindRelationship' · RELCHAIN-NAME · SUB
   Rel-Rind = SUB · 'rindRelationship' · NUM · SUB
       Cons = Val ('<' | '>' | '≤' | '≥' | '=' | '≠') Val
       Prin = SUB | OBJ
        Val = NAME | VAR | NUM
```

NAMEs start with a lowercase letter and can contain letters, numbers and underscores, while VARs start with a uppercase letter. SUB and OBJ is a NAME or VAR that identifies a subject or an object. SF and PIF are the *sensitivity flag* and *primary instance flag*, respectively. These are used during the SocACL negotiation process, which is not covered in this paper.

4 SocACL Example

Suppose we have some hypothetical OSN with a member Alice that has a coworker Bob (eq. (1)) and considers this relationship non-sensitive. She is envious of people enrolled at the prestigious University of Learning (eq. (2)), treating this attribute as non-sensitive and a non-primary instance.

$$\text{alice } \mathbf{says\ Me} \cdot \mathbf{relationship} \cdot \text{coworker} \cdot \text{bob } \mathbf{: ns;} \tag{1}$$

$$\text{alice } \mathbf{says\ Me} \cdot \text{envious} \cdot \mathbf{Other : ns \cdot np\ if}\ \mathbf{Other} \cdot \text{enrolled} \cdot \text{“UoL”}\mathbf{;} \tag{2}$$

Below we find the ASP translation of eq. (1) and (2) respectively. For eq. (3) and (4) arity 1 denotes the principal providing this attribute. In eq. (4) arity 1, 4, and 5 of “enrolled” are underscores, the anonymous variable of DLV used as a placeholder for values that do not matter for this rule. Meaning that for “enrolled” it does not matter who provides this attribute (arity 1). Arities 4 and 5 are the SF and PIF respectively, since these are used only by the SocACL negotiation process it does not matter what their values are when used as decision criteria.

$$\mathbf{relationship}(\text{alice}, \text{alice}, \text{bob}, \text{coworker}, \mathbf{ns}). \tag{3}$$

$$\text{envious}(\text{alice}, \text{alice}, \mathbf{Other}, \mathbf{ns}, \mathbf{np}) \leftarrow \text{enrolled}(\_, \mathbf{Other}, \text{“UoL”}, \_, \_). \tag{4}$$

5 Related Work and Conclusion

With relationships an integral part of any OSN there have been various access control framework proposals based on them. *Relationship-Based Access Control* (ReBAC) [2] and its supporting language specifies policies in terms of the accessors relationship with the owner. ReBAC's modelling of relationships differs from that of SocACL. ReBAC relationships can be composed from “smaller” relationships, e.g. “grandparent” can be composed from “parent parent”, which can also be inverted. With SocACL allowing for distance based relationships these compositions pose a problem; is “grandparent” a 1st- or 2nd-degree relationship? Instead, SocACL allows for indirect relationships to be expressed as a sequence of direct relationships at each “hop”. Furthermore, SocACL relationships can be used in conjunction with attributes and the aggregate operations count, sum, min and max. This allows for rules such as *Allow access to “friends”, with red hair, which have 5 “friends” in common with me*; something not possible with ReBAC.

In SocACL we have an access control language with features tailored to OSNs. SocACL utilises the two most abundant sources of information in OSNs as decision criteria; information about the user and their relationships with others.

References

1. Baral, C.: Knowledge Representation, Reasoning and Declartive Problem Solving, 1st edn. Cambridge University Press (2010)
2. Fong, P.W.L.: Relationship-based access control: protection model and policy language. In: Proc. of the 1st ACM Conf. on Data and Application Security and Privacy, CODASPY 2011, pp. 191–202. ACM, New York (2011)
3. Gao, H., Hu, J., Huang, T., Wang, J., Chen, Y.: Security Issues in Online Social Networks. IEEE Internet Computing 15(4), 56–63 (2011)
4. Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The dlv system for knowledge representation and reasoning. ACM Trans. Comput. Logic 7(3), 499–562 (2006)
5. Madejski, M., Johnson, M., Bellovin, S.M.: A Study of Privacy Settings Errors in an Online Social Network. In: Proc. of 2012 IEEE Int. Conf. on Pervasive Computing and Communications Workshops (PERCOM Workshops), pp. 340–345 (March 2012)